p-75

NASA Technical Memorandum 104161

COMPUTER PROGRAMS FOR THE CALCULATION OF DUAL STING PITCH AND ROLL ANGLES REQUIRED FOR AN ARTICULATED STING TO OBTAIN ANGLES OF ATTACK AND SIDESLIP ON WIND-TUNNEL MODELS

John B. Peterson, Jr.

October 1991

(NASA-TM-104161) COMPUTER PROGRAMS FOR THE CALCULATION OF DUAL STING PITCH AND ROLL ANGLES REQUIRED FOR AN ARTICULATED STING TO OBTAIN ANGLES OF ATTACK AND SIDESLIP ON WIND-TUNNEL MODELS (NASA) 75 p CSCL 14B G3/09

N92-12007

Unclas 0052280

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

INTRODUCTION

In many wind tunnels, the model support sting drive can both pitch and roll the sting support. In addition, some tunnels have articulated stings that are hinged between the sting drive mechanism and the model to pitch the model to higher angles. This allows tests to be made at higher angles of attack than those available with the regular sting drive.

Determination of the angles necessary to set the sting drive mechanism and the articulated sting pitch angle is simple when only a given angle of attack is desired. However, if an angle of sideslip is desired, it can be very difficult to determine the sting angles necessary to give the desired α and β on the model.

A program has been developed (ref. 1) to compute the pitch and roll of a conventional wind tunnel sting to position a model at the desired α and β. This program accepts stings with offset angles in yaw, pitch, and roll.

Also, an algorithm is described in reference 2 that uses matrix algebra to develop equations for the sting drive angles necessary to position a wind tunnel model in a wind tunnel. These equations can solve for three degree of freedom sting drive angles necessary to position the model at any attitude including the effects of sting offsets and sting bending angles.

The programs in this report compute the pitch and roll position of the conventional sting drive and the pitch of the high angle articulated sting to position the model at the desired α and β and position the model as near as possible to the centerline of the tunnel.

Two computer programs have been developed and are described in this report. These programs cover the case of no accelerometers on board the model and the case of accelerometers on board the model to measure model pitch and/or model roll with respect to gravity.

These programs are iterative programs in that they calculate the α and β position of the model in the tunnel based on assumed pitch and roll angles of the sting drive and the pitch of the articulated sting; and then calculate new positions for the sting pitch and roll and second or articulated pitch to move the model closer to the desired α and β, and position the model in the center of the tunnel. This process is continued until the calculated α and β of the model match the desired α and β within a small increment and the model is close as possible to the center of the tunnel. This procedure insures that the α and β calculated by the data reduction program based on the available information such as stream flow angles, sting bending due to measured model forces, and on board measurements of model pitch and roll, if available, will match the desired α and β.

Both of these programs accept three sting offset angles, three sting bending angles, and two tunnel flow angles. In addition, the second program accepts on board measured pitch angle and on board measured roll angle, if available.

SYMBOLS

u	Velocity along X axis of model
v	Velocity along Y axis of model
w	Velocity along Z axis of model
X	Longitudinal body axis, positive aft, see figure 1
Y	Lateral body axis, positive to right, see figure 1
Z	Vertical body axis, positive upward, see figure 1
α	Model angle of attack, $\alpha = \arctan(w/u)$, see figure 1
β	Model angle of slideslip, $\beta = \arcsin(-v/V_{\infty})$, see figure 1
θ	Model pitch angle, positive direction is nose up, see figure 1
θ_{off}	Sting offset angle in pitch
θ_{sb}	Sting bending in pitch
θ_{sc}	Sting drive pitch command
θ_{sc2}	Articulated sting pitch command
ϕ	Model roll angle, positive direction is right wing down, see figure 1
ϕ_{off}	Sting offset angle in roll
ϕ_{sb}	Sting bending in roll
ϕ_{sc}	Sting drive roll command
ϕ_{sc2}	Articulated sting roll
ψ	Model yaw angle, positive direction is nose right, see figure 1
ψ_{off}	Sting offset angle in yaw
ψ_{sb}	Sting bending in yaw

DETERMINATION OF ANGLE OF ATTACK AND ANGLE OF SIDESLIP FOR A WIND TUNNEL MODEL

The definition of angle of attack, α, on a wind-tunnel model is given in reference 3 as the arctan (w/u) where w is the component of the free stream velocity along the Z axis of the model (vertical axis with the positive direction upward) and u is the component of the free stream velocity along the X axis of the model (longitudinal axis with the positive direction aft). This definition applies no matter what the orientation of the model is. By examining the signs of u and w, the correct quadrant for α can be determined between -180° and +180°. If both u and w are zero, then the angle of attack is indeterminate, but in these programs, when both u and w are equal to zero, α is defined to be equal to zero

The angle of sideslip, β, is defined in reference 3 as the arcsin ($-v/V_{\infty}$) where v is the component of the free stream velocity along the Y axis of the model (the lateral axis with the positive direction out the right or starboard wing) and V_{∞} is the total free stream velocity. This means that β is positive when the flow is from the right.

In these computer programs, the free stream velocity, V_{∞}, is set equal to one, and the components of the free stream velocity along the wind-tunnel axis system (u, v, and w) are calculated from the angles of upwash and sidewash in the wind tunnel. The upwash angle (UWA) is defined to be positive when the flow is upward in the tunnel (i.e., the w component is positive) and the sidewash to be positive when the flow is from the right to the left (from the starboard to the port). For positive wind-tunnel sidewash angle and the model pitch, roll and yaw angles equal to zero, the v component of the model velocity is negative and the model sideslip angle, β, is positive.

The three velocities (u, v, and w) along with the three axes of the model are recalculated for each rotation of the model. By convention, the rotations are taken in order of yaw, pitch, and then roll (i.e., rotation about the Z axis, and Y axis, and then the X axis of the model). The equations for the u, v, and w velocities after rotation through each of the angles are given below.

Yaw (rotation about Z axis), ψ

$$UA = U * \cos(\psi) - V * \sin(\psi)$$

$$VA = V * \cos(\psi) + U * \sin(\psi)$$

$$WA = W$$

Pitch (rotation about Y axis), θ

$$UB = UA * \cos(\theta) - WA * \sin(\theta)$$

$$VB = VA$$

$$WB = WA * \cos(\theta) + UA * \sin(\theta)$$

Roll (rotation about X axis), ϕ

$$UC = UB$$

$$VC = VB * \cos(\phi) - WB * \sin(\phi)$$

$$WC = WB * \cos(\phi) + VB * \sin(\phi)$$

Where ψ is the angle of yaw (positive for nose right), θ is the angle of pitch (positive for nose up), and ϕ is the angle of roll (positive for right wing down). These angles are shown in figure 1 from reference 4.

The equations given above are used in subroutine ALPBET of program

STNG2PR to calculate the velocities u, v, and w after each rotation of the model. After the three velocities are determined, the angle of attack, α, the angle of sideslip, β, are calculated using the formulas given above.

DESCRIPTION OF COMPUTER PROGRAMS

Two computer programs were developed to calculate the sting pitch angle, θ_{sc}, sting roll angle, ϕ_{sc}, and the second sting pitch angle, θ_{sc2}, necessary to obtain the desired α and β and to position the model as close as possible to the center line of the tunnel. Since this problem has three dependent variables (θ_{sc}, ϕ_{sc}, and θ_{sc2}), it requires that three independent quantities be specified in order that the problem be completely defined. The variables chosen are the α and β of the model and the position of the model in the tunnel with respect to the centerline of the tunnel. When no restraints are given for the three dependent variables (θ_{sc}, ϕ_{sc}, and θ_{sc2}), the program will find a solution that gives the desired α and β and puts the model on the centerline of the tunnel. However, this solution may require sting pitch angles, θ_{sc}, greater than or less than those obtainable from the tunnel sting pitch drive (see figure 2). In these cases, the program puts the sting at the limit of the sting drive and changes the other two variables (ϕ_{sc} and θ_{sc2}) to position the model at the desired α and β and as close as possible to the tunnel centerline.

The first program, STNG2, calculates the sting pitch and roll and second sting pitch angles based on the sting geometry, such as sting offsets and sting bending. This program consists of the main program and three subroutines. A listing of the program is given in Appendix A.

The second program, STNG2PR, calculates the sting pitch and roll and second sting pitch angles from the sting geometry but also uses inputs from accelerometers on board the model that measure the model pitch and roll angles. This program consists of a main program and five subroutines and a listing is given in Appendix B.

A list and description of the variables used in these programs is given after the description of the programs.

PROGRAM STNG2

Program STNG2 calculates the sting pitch, θ_{sc}, sting roll, ϕ_{sc}, and second sting pitch, θ_{sc2}, necessary to obtain a desired model angle of attack, α, angle of sideslip, β, and position the model in the center of the tunnel. The program actually first calculates the α, β, and position of the model in the tunnel for an assumed sting pitch, sting roll, and second sting pitch. Then, by iteration, the assumed sting pitch, sting roll, and second sting pitch angles are changed until the calculated α and β agree with the desired or command α and β (ALPC and BETC) and the model is located on the centerline of the tunnel.

Program STNG2 is very similar to program STNG reported in reference 1. Program STNG2 has three factors, called FA, FB, and FD, which are equal to the change in alpha, beta, and distance from the centerline of the tunnel, respectively, with a unit change in the angle of sting (either pitch, θ_{sc}, roll ϕ_s, or second pitch angle, θ_{sc2}, depending on which angle is being optimized.) These factors are combined linearly with the error between the actual value of the angle or distance (ALP, BET, and DIS) and the desired angle or distance (ALPC, BETC, and zero distance) to give the function F. This function is then minimized by subroutine CONV which uses a procedure which is equivalent to the secant method of finding the roots of an equation to find the sting position that reduces F to a minimum value.

Each time subroutine CONV is called to find the sting position to minimize F, subroutine VEL is called to determine the α and β, and subroutine DIS is called to determine the distance of the model from the centerline, at the new sting position. These new values for α, β, and distance are then used to calculate a new value for F. This process is repeated until the absolute value of F is less than the tolerance for F (TOLF = 0.00001) or for a maximum of three iterations.

The function F is calculated and minimized first for sting pitch (THESC), then for sting roll (PHISC), and then for the second sting pitch (THESC2).

After the F has been minimized for each of the three sting angles, the entire procedure is repeated until the change in each of the three sting angles (THESC, PHISC, and THESC2) is less than the tolerance for sting angles (TOLANG = 0.01) in the last iteration or for a maximum of 100 iterations.

The program usually reaches the limit for TOLANG in about 50 iterations with the α and β within about 0.001 degrees of the desired angle and the distance equal to about 0.2 (where the units for distance are the same as used to specify the sting lengths, R1, X2, and X3 at the beginning of the program).

The program then checks to see if the sting roll position is outside the limits set for the sting roll position at the beginning of the program (-85° to 100° for model upright).

If the sting roll is outside the limits, the program is rerun from statement 50 with the pitch equal to negative of the pitch solution and the roll equal roll solution $\pm$ 180°. The solution with the new initial values for pitch and roll is very quick and the sting roll will be between the limits of -85° to 100°. If an inverted solution is desired, the limits on the sting roll can be changed to +80° for the lower limit (PHISLL) and 265° for the upper limit (PHISLU).

The program then checks to see if the sting pitch (THESC) is within the limits for sting pitch available in the wind tunnel. The lower limit, THESLL, is set to -11° and the upper limit, THESLU, is set to +19° in this program. If the sting pitch command THESC is within the sting limits, the program continues on to the section of the program starting at the DO 2000 statement where the angle of attack, ALP, and the angle of sideslip, BET, are converged to the commanded angle of attack, ALPC, and the commanded angle of sideslip, BETC, without regard to the distance of the wind tunnel model from the centerline of the tunnel. In this section of the program, the factor F is a function of the error in angle of attack, DALP and angle of sideslip, DBET only. Therefore,

$$F = FA*DALP + FB*DBET$$

As before, the function F is calculated and minimized first for the sting pitch (THESC), then for the sting roll (PHISC), and then for the second sting pitch (THESC2). This procedure is repeated until the error in angle of attack, DALP, and the error in angle of sideslip, DBET is reduced to less than the tolerance TOLDAB. (Set to 0.00001 at the beginning of the program).

If the sting pitch command, THESC, is outside the limits for sting pitch available in the wind tunnel, the sting pitch is set to the nearest limit and the program goes to statement 2020. In this part of the program as in the part after the DO 2000 statement, the program converges the angle of attack, ALP, and angle of sideslip, BET, to the commanded angles of attack and sideslip without regard to the distance from the wind tunnel model to the centerline of the tunnel. However, the sting pitch is fixed at the sting pitch limit and only the second sting pitch, THESC2, and the sting roll, PHISC, are used (in that order) to reduce the error in ALP and BET.

There is a counter in the program (IREPT) which is used to stop execution if the program gets into an infinite loop when it returns to statement 50 over and over. This can happen if the program is asked to solve an impossible case such as the BETC of 20° when the sting pitch limits are +19° and -11°.

Since STNG2 converges much more slowly than the STNG program did (sometimes 65 iterations in STNG2 compared to generally 10 to 20 in STNG), the answers from the previous calculation are saved and used as a starting point for the next calculations in the STNG2 program.

The answers for the first part of the program where THESC is unlimited are saved as the variables THESCUL, PHISCUL, and THES2UL. The answers of the last part of the program where THESC is limited are saved as the variable THESC2L and PHISCL.

The following is a list and description of the variables used in program STNG2 (all angles are degrees):

ALP	Angle of attack of model, α
ALPC	Command or desired angle of attack
BET	Angle of sideslip of model, β
BETC	Command or desired angle of sideslip
DALP	Differences between angle of attack of the model and desired angle of attack
DBET	Difference between angle of sideslip of the model and desired angle of sideslip
DIS	Distance of the wind tunnel model from the centerline of the wind tunnel measured perpendicular to the wing span (with $\phi_{sc2} = 0°$).
DY	Lateral distance of the wind tunnel model from the centerline (Positive to the right)
DZ	Vertical distance of the wind tunnel model from the centerline (Positive upward)
F	Function which is to be minimized by pitching and rolling the sting drive and pitching the high angle articulated sting
FA	Weighting factor for DALP in the function F
FB	Weighting factor for DBET in the function F
FD	Weighting factor for DIS in the function F
IREPT	Variable used to limit the number of times the program starts over after converging outside the sting roll limits
PHIOFF	Sting offset angle in roll (ϕ_{off})
PHISB	Sting bending in roll ϕ_{sb})
PHISC	Sting roll command (ϕ_{sc}) required to position the model at the

desired α and β and at the centerline of the tunnel

PHISCL	Sting roll command with sting pitch fixed at the limit. Used as a starting point for the next computation after statement 2020
PHISCUL	Sting roll command with sting pitch unlimited. Used as a starting point for the next computation just above statement 50
PHISC2	Second (articulated) sting offset in roll. Set to a constant value at the beginning of the program and not changed by the program.
PHISLL	Lower limit on sting roll.
PHISLU	Upper limit on sting roll.
PSIOFF	Sting offset angle in yaw (ψ_{off}).
PSISB	Sting bending in yaw (ψ_{sb}).
R1	Radius of the sting pitch arc sector (see figure 2).
SWA	Wind tunnel free stream sidewash angle, positive for flow from right.
THEOFF	Sting offset angle in pitch (θ_{off}).
THESB	Sting bending angle in pitch (θ_{sb}).
THESC	Sting pitch command (θ_{sc}) required to position the model at the desired α and β.
THESCUL	Sting pitch command with sting pitch unlimited. Used as a starting point for the next computation just above statement 50.
THESC2	Second (articulated) sting pitch angle.
THESC2L	Second sting pitch angle with sting pitch fixed at the limit. Used as a starting point for the next computation after statement 2020.
THESLL	Lower limit for sting pitch angle.
THESLU	Upper limit for sting pitch angle.
THES2UL	Second sting pitch angle with sting pitch unlimited. Used as a starting point for the next computation just above statement 50.
TOLANG	Convergence tolerance for the movement of the three sting angles

during the last iteration.

TOLDAB	Convergence tolerance for the sum of the absolute values of DALP and DBET.
TOLF	Convergence tolerance for the value of F.
UWA	Wind tunnel free stream upwash angle, positive for flow from below.
u	Free stream velocity component in the longitudinal direction in the wind tunnel (the total free stream velocity is assumed to be 1.0).
v	Free stream velocity component in the lateral direction (flow from the right when looking forward is positive).
w	Free stream velocity component in the vertical direction (upward flow is positive).
X2	Distance from sting pitch arc sector to second (articulated) sting pitch axis. (See figure 2.)
X3	Distance from second sting pitch axis to wind tunnel model. (See figure 2.)

SUBROUTINE VEL

The purpose of subroutine VEL is to calculate the angle of attack, α, and angle of sideslip, β, of the wind tunnel model. The subroutine requires inputs of the velocity components in the wind tunnel u, v, and w; the sting offset angles; the sting bending angles; and the sting drive angles. The subroutine calculates the components of the free stream velocity along the three axes of the model after each rotation angle using the formulas given in the section "Determination of Angle of Attack and Angle of Sideslip for a Wind Tunnel Model." After the last rotation, these velocities are used to calculate the angle of attack, α, and angle of sideslip, β, using the following formulas:

$$\alpha = \arctan (w/u)$$

$$\beta = \arcsin (-v/V_{\infty})$$

V_{∞} is set equal to one in the main program (STNG2) and therefore:

$$\beta = \arcsin (-v)$$

The following is a list and description of the additional variables used in subroutine VEL:

UB, ... ,UI	Longitudinal velocity in the model axis system after each rotation.
VB, ... ,VI	Lateral velocity in the model axis system after each rotation.
WB, ... ,WI	Vertical velocity in the model axis system after each rotation.

SUBROUTINE DISTANCE

Subroutine distance calculates the position of the wind tunnel model relative to the centerline of the wind tunnel. The subroutine requires inputs of the lengths of the various portions of the sting, R1, X2, and X3; the sting offset angles in yaw (PSIOFF), pitch (THEOFF) and roll (PSIOFF); and the sting drive angles, THESC, PHISC, and THESC2. The distance DY and DZ are calculated by subroutine DIST where DY is the lateral distance of the model from the wind tunnel centerline with positive direction to the right and DZ is the vertical distance from the centerline with positive values above the centerline. The units of the distances DY and DZ are the same as the units used to input the values of R1, X2, and X3.

The program starts with an axis system aligned to the model axis system and then translates and rotates the axis system along the stings until it is aligned to the wind tunnel axis system. As each of these translations and rotations are made, the subroutine calculates the position of the model in the axis system after each movement. Since the translations and rotations are taken in reverse order (normally the program calculates the effect of sting pitch first, and second sting roll last, (i.e., see subroutine VEL). The angles used in subroutine DIST are the negative of the angles used in the normal procedure.

The following is a list and description of the additional variables used in subroutine DIST:

XA, ..., XJ	Longitudinal distance of the wind tunnel model in the translated and/or rotated axis system (positive downstream).
YA, ..., YJ	Lateral distance of the wind tunnel model in the translated and/or rotated axis system (positive to the right facing upstream).
XA, ..., ZJ	Vertical distance of the wind tunnel model in the translated and/or rotated axis system (positive upward).

SUBROUTINE CONV

The purpose of subroutine CONV is to minimize a function Y = F(X). The function is calculated in the calling program and the subroutine calculates a new value of the independent variable X that will make the value of the dependent variable Y, nearer to zero. The new value of X is determined by calculating where a straight line through the last two previous pairs of points; X and Y, and XSAVE and YSAVE; intersects the X axis.

This new value is returned to the calling program in the X parameter and the old X and Y are placed in the XSAVE and YSAVE parameters at the end of the subroutine. In order to improve the stability of this procedure, certain limits are placed on the distance the new X value can be from the old X and XSAVE values. The IF statement above statement 3 of the subroutine CONV limits the new X value to be no more than 3.5 times the distance between the old X and XSAVE values away from the average of these values. Also if the two previous values for X are the same, the new X is calculated in a special way in statement 3. If the two previous Y values are the same, the new X is calculated to be the average of the two previous X's in statement 4.

The following is a list and description of the variables used in subroutine CONV:

DX	Absolute distance between X1 and X2.
X	Latest value of the independent variable. Also returned as the next value to be tried for the independent variable.
XA	Average of X1 and X2.
XK	Slope of line between X1, Y1 and X2, Y2.
XSAVE	Previous value of independent variable. Also returned as previous value of X.
X1	Previous value of the independent variable (same as XSAVE).
X2	Latest value of the independent variable.
Y	Latest value of the dependent variable.
YSAVE	Previous value of the dependent variable. Also returned as the previous value of Y.
Y1	Previous value of the dependent variable (same as YSAVE).
Y2	Latest value of the dependent variable.

PROGRAM STNG2PR

Program STNG2PR calculates the sting pitch and roll angles and articulated sting pitch angle required to obtain the desired angle of attack and angle of sideslip and position the model near the centerline of the wind tunnel when the model has on board measurements of the model pitch and roll (or pitch only if the roll is not available). The model pitch and roll is assumed to be measured relative to gravity.

At first it would appear that these cases would simplify the calculation of the alpha and beta of the model in the wind tunnel since the effects of sting offsets and sting bending are already included in the on board measurements of model pitch and roll. This is true in the case where a straight sting is used and the model yaw is assumed to equal zero. Since most wind tunnel tests use straight stings, on board measurements of model pitch and roll is a very valuable method of determining model alpha and beta in the wind tunnel. For bent stings, however, model yaw cannot be assumed to equal zero and model pitch and roll alone is not sufficient to determine model angle of attack and angle of sideslip.

In order to make program STNG2PR more generally applicable, no assumptions were made about the sting offset angles or the model yaw angle in the wind tunnel. Therefore, the program is applicable not only to the case of straight sting, but also to the case where the sting is offset and/or where sting bending occurs.

Program STNG2PR is similar to program STNG2 in that they both calculate the model α, β, and position of the model in the tunnel at a given sting position and then, by the use of a factor, try to reduce the difference between the calculated α and β and the desired or commanded α and β and position the model near the centerline of the wind tunnel.

Program STNG2PR uses the same function (F) as program STNG2 uses to reduce the error between the calculated α and β and the desired α and β and to position the model near the centerline of the wind tunnel. The major difference between the two programs is that STNG2PR uses a three-step process to calculate α and β instead of the one step that STNG uses (subroutine VEL). The three steps are: first, calculate the yaw, pitch and roll angles of the model from the sting support system angles and the sting geometry (subroutine SIMUST), second, correct the pitch and roll of the model by the difference between the measured model pitch and roll, and the calculated model pitch and roll (DTHEMOB and DPHIMOB) determined at the beginning of the program; and third, calculate the α and β of the model from the model yaw, pitch and roll angles (subroutine ALPBET).

When programs STNG2 and STNG2PR are used in actual wind tunnel situations, the sting drive angles calculated by these programs will change somewhat as the sting and model move to the commanded positions because the sting bending angles will change as the angles of attack and sideslip of the model change, and because the differences between the calculated model pitch and roll and the on board measured model pitch and roll change as the model attitude changes. The final position, however, will be the correct position to obtain the desired α and β, since the final values for the sting bending and measured model pitch and roll will be the same as those used by the wind tunnel data reduction program.

The following is a list and description of the additional variables used in program STNG2PR (all angles are in degrees):

DPHIMOB	Difference between PHIMOB and PHIMT determined at the beginning of the program.
DTHEMOB	Difference between THEMOB and THEMT determined at the beginning of the program.
PHIMOB	Roll and model as measured by on board accelerometers.
PHIMOBT	Theoretical on board roll of model. PHIMOBT = PHIMT + DPHIMOB
PHIMT	Theoretical roll of model as determined by subroutine SIMUST from the sting drive angles, sting offset angles, and sting bending angles.
PHIS	Actual sting drive roll angle.
PHIS2	Actual articulated sting pitch angle.
PSIMT	Theoretical yaw angle of the model as determined by subroutine SIMUST from the sting drive angles, sting offset angles, and sting bending angles.
THEMOB	Pitch of model as measured by on board accelerometers.
THEMOBT	Theoretical on board pitch of model. THEMOBT = THEMT + DTHEMOB
THEMT	Theoretical pitch of model as determined by subroutine SIMUST from the sting drive angles, sting offset angles and sting bending angles.
THES	Actual sting drive pitch angle.
THES2	Actual articulated sting pitch angle.

SUBROUTINE SIMUST

Subroutine SIMUST simulates the sting-support, sting, and model system mathematically to calculate the model yaw, pitch and roll angles from the sting drive, sting offset and sting bending angles. The method of calculating the model yaw, pitch and roll is discussed in reference 4 and 5, and is explained below.

The pitch angle of the model is determined by calculating the X component, in the model axis system, of a unit vector in the Z direction of the tunnel axis system (XZ). The pitch angle is then the arcsin (-XZ). The pitch angle can range from -90° to +90°.

The roll of the model is determined by calculating the Y and Z components, in the model axis system, of a unit vector in the Z direction of the tunnel axis system (YZ and ZZ). The roll of the model is then the arctan (-YZ/ZZ) where the quadrant of the roll angle is determined by the sings of YZ and ZZ individually. The roll angle can range from -180° to 180°. If both YZ and ZZ are zero (i.e., the pitch angle is ±90°) then the roll of the model is determined by the arctan (-YX, ZX) and the yaw of the model is defined to be equal to zero (where YX and ZX are the Y component and the Z component respectively of a unit vector in the X direction of the tunnel axis system).

The yaw of the model is determined by calculating the X component in the model axis system of a unit Y vector in the wind tunnel axis system, XY, and the X component in the model axis system of a unit X vector in the tunnel axis system, XX. The yaw of the model is then the arctan (-XY/XX) and can range from -180° to 180°.

The following is a list and description of the additional variables used in subroutine SIMUST:

PHIS	Sting drive roll angle.
PHIS2	Second sting roll angle.
THES	Sting drive pitch angle.
THES2	Second sting drive pitch angle.
XX, YX, ZX	The X, Y, and Z components in the model axis system of a unit vector in the X direction in the tunnel axis system.
XY, YY, ZY	The X, Y, and Z components in the model axis system of a unit vector in the Y direction in the tunnel axis system.
XZ, YZ, ZZ	The X, Y, and Z components in the model axis system of a unit vector in the Z direction in the tunnel axis system.

SUBROUTINE COMP

Subroutine COMP is used to calculate the components, in the model axis system, of a unit vector in the wind tunnel axis system so that the model yaw, pitch and roll angles can be determined. This subroutine is very similar to subroutine VEL in program STNG2 which was used to calculate the velocity components in the model body axis system. Although this subroutine can calculate the X, Y, and Z components, in the model axis system, of an arbitrary vector in the wind tunnel axis system, it is only used in this program to calculate the components of a unit vector in the X, Y, or Z direction in the wind tunnel axis system. This means that one of the components, X, Y, or Z, is set equal to one and the other components are set equal to zero in the calling program argument list. The components in the model body axis system of the unit vectors are then used in subroutine SIMUST to calculate the model yaw, pitch, and roll.

The following is a list and description of the additional variables used in subroutine COMP:

X, Y, Z	X, Y, and Z components of a vector in the tunnel axis system.
XB,...,XI	X component in the model axis system of a vector in the tunnel axis system after each rotation.
YB,...,YI	Y component in the model axis system of a vector in the tunnel axis system after each rotation.
ZB,...,ZI	Z component in the model axis system of a vector in tunnel axis system after each rotation.

SUBROUTINE ALPBET

The purpose of subroutine ALPBET is to calculate the angle of attack, α, and angle of sideslip, β, of the wind tunnel model. The subroutine requires inputs of the velocity components in the wind tunnel, U, V, and W, and the model Euler angles, yaw (ψ), pitch (θ), and roll (ϕ). The subroutine calculates the components of the free stream velocity along the three axes of the model after each rotation angle using the formulas given in the section "Determination of Angle of Attack and Angle of Sideslip for a Wind Tunnel Model." After rotation through the three Euler angles, the velocities are used to calculate the angle of attack, α, and angle of sideslip, β, using the following formulas:

$$\alpha = \arctan (v/u)$$

$$\beta = \arcsin (-v/V_\infty)$$

V_∞ is set to one in the main program (STNG2PR) and therefore:

$$\beta = \arcsin (-v)$$

The following is a list and description of the additional variables used in subroutine ALPBET (all angles are in degrees):

ALP	Angle of attack of model, α.
BET	Angle of sideslip of model, β.
PHIM	Angle of roll of model, ϕ.
PSIM	Angle of yaw of model, ψ.
THEM	Angle of pitch of model, θ.
UA, UB, UC	Longitudinal velocity component in the model axis system after each Euler angle rotation.
VA, VB, VC	Lateral velocity component in the model axis system after each Euler angle rotation.
WA, WB, WC	Vertical velocity component in the model axis system after each Euler angle rotation.

CONCLUDING REMARKS

Two programs have been developed to calculate the pitch, roll, and second articulated pitch angles of a wind tunnel sting drive system that will position a model near the centerline and at the desired angle of attack and angle of sideslip in the wind tunnel. These programs account for the effects of sting offset angles, sting bending angles and wind tunnel stream flow angles. In addition, the second program incorporates inputs from on board accelerometers that measure model pitch and roll with respect to gravity.

These program solve for the desired sting pitch and roll with an interactive procedure using the forward equations that calculate the model position, α and β from the sting geometry and the sting pitch and roll. This procedure allows sting offset angles, sting bending angles, and stream flow angles to be taken into account.

A copy of the source code of these two programs can be obtained from the Langley Computer Center with the following statements:

GET, ZSTNG2/UN = 003101N

or GET, ZTNG2PR/UN = 003101N

The run times for these programs very depending upon the number of iterations required to converge to a solution. The programs were compiled and run under Fortran 5 (Fortran 77) on the Control Data Corporation Cyber CY180-860 computer at Langley Research Center. The run times for STNG2 are from 0.065 seconds for one iteration to 36.4 seconds for 100 iterations and 6 restarted calculations (the maximum allowed in these programs).

Normally, however, run time for STNG2 vary from about 1.8 seconds for large changes in α and β from the previous computation to 0.38 second for small changes in α and β (i.e., 2°). Very small changes (i.e., 0.01°) require only two iterations and 0.1 second run time.

Run times for STNG2PR are about twice as long as the run times for STNG2.

REFERENCES

1. Peterson, John B., Jr., Computer Programs for Calculation of Sting Pitch and Roll Angles Required to Obtain Angles of Attack and Sideslip on Wind Tunnel Models, NASA TM 100659, July 1988.

2. Heim, E. R. and Hobbs, R. W., Development of a Versatile Rotation Transformation Algorithm for Automatic Model Attitude Positioning. Journal of Aircraft, Vol. 26, No. 8, August 1989, pp. 718 - 722.

3. Letter Symbols for Aeronautical Sciences. ASA Y10.7-1954, The American Society of Mechanical Engineers, 154.

4. Foster, Jean M. and Adcock, Jerry B., Users Guide for the National Transonic Facility Data System, NASA TM-100511, December 1987.

5. Fox, Charles H., Jr., Real Time Reduction Capabilities at the Langley 7- by 10-Foot High Speed Tunnel, NASA TM-78801, January 1980.

APPENDIX A

COMPUTER LISTING OF PROGRAM STNG2

This appendix contains a computer listing of the program STNG2 which calculates the wind tunnel sting pitch and roll angles and second sting pitch angle required to obtain a desired angle of attack, α, and sideslip, β, on a wind tunnel model and to position the model in the center of the tunnel. The program accepts inputs of stream flow angles in two directions and sting offsets and sting bending angles in three directions

```
      PROGRAM STNG2 (INPUT,OUTPUT)
C THIS PROGRAM CALCULATES THE STING ANGLES FOR THE HIGH ALPHA STING SYSTEM
C TO GET THE DESIRED ALPHA AND BETA AND THE MINIMUM DISTANCE FROM THE
C CENTERLINE OF THE TUNNEL  ( ONLY THE SECOND PITCH ANGLE IS USED, NOT
C THE SECOND ROLL ANGLE)
C
C ANSWERS SAVED FOR NEXT COMPUTATION
C
C THE ORDER OF ROTATIONS ARE : STING PITCH (THESC), STING ROLL (PHISC),
C OFFSET YAW (PSIOFF), OFFSET PITCH (THEOFF), OFFSET ROLL (PHIOFF),
C SECOND STING PITCH (THESC2), SECOND STING ROLL (PHISC2),
C BENDING IN YAW (PSISB), BENDING IN PITCH (THESB), AND BENDING IN ROLL (PHIS
C
C CODED BY -- JOHN B. PETERSON, JR.   NASA/LARC/AAD/HRNAB   1990
C

C          STING OFFSETS
      PSIOFF=0.
      THEOFF=-20.
      PHIOFF=0.
      PHISC2=0.

C          STING LENGTHS  (SHOULD BE ON THE ORDER OF 57.3 UNITS SO THAT
C          DISTANCES AND ANGLES ARE OF SAME ORDER OF MAGNITUDE)
C          R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
C          X2 IS THE LENGTH OF THE MAIN STING.
C          X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
C
      R1=122.5
      X2=87.5
      X3=48.918

C          LIMITS

C          LIMITS ON STING ROLL ANGLE
      PHISLL=-85.
```

```
      PHISLU=100.
C FOR INVERTED RUNS USE FOLLOWING LIMITS
C     PHISLL=80.
C     PHISLU=265.

C               LIMITS ON STING PITCH ANGLE (THESE LIMITS ALSO LIMIT
C               THE AVAILABLE BETA TO +19 & -11 UPRIGHT AND -19 & +11 INVERTED)
      THESLL=-11.
      THESLU=19.

C         STREAM FLOW ANGLES
C           POSITIVE FOR FLOW FROM BELOW AND FROM RIGHT
      SWA=.0
      UWA=.0
C               FREE STREAM VELOCITIES (TOTAL VEL. = 1.0)
      U=SQRT(  1./ (1.+(TAND(SWA))**2+(TAND(UWA))**2 )  )
      V=-U*TAND(SWA)
      W= U*TAND(UWA)

C               SET INITIAL VALUES FOR VARIABLES

      THESCUL=0.0
      PHISCUL=0.0
      THES2UL=20.

      PHISCL=0.0
      THESC2L=20.

      THESSAV=999.
      PHISSAV=999.
      THES2SV=999.

C               TOLERANCES

      TOLF=.00001
      TOLDAB=.00001
      TOLANG=.01
```

```
C                          INPUTS TO CONTROL PROGRAM

C                   COMMAND ANGLES
      ALPC=40.
      BETC=10.

C                   STING DEFLECTIONS
      PSISB=0.0
      THESB=0.0
      PHISB=0.0

C                          END OF INPUTS TO CONTROL PROGRAM

C                  SET TO PREVIOUS ANSWERS UNLIMITED IN THESC
       THESC  = THESCUL
       PHISC  = PHISCUL
       THESC2 = THES2UL

C                  SET REPEAT COUNTER TO 0
      IREPT = 0

   50 CONTINUE

      IREPT = IREPT + 1

         PRINT 98
         PRINT 96
         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
     *            PHIOFF

C                       CONVERGE ON ALPHA COMMAND (ALPC), BETA COMMAND (BETC),
C                       AND DISTANCE (DIS)

      DO 1000 ICONV=1,100

      THESCSV=THESC-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESCSV,PHISC,THESC2,PHISC2,ALPSV,BETSV)
```

```
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESCSV,PHISC,THESC2,
     *          DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC   ,PHISC,THESC2,PHISC2,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC   ,PHISC,THESC2,
     *          DY   ,DZ   )
      DALP  =ALP   - ALPC
      DBET  =BET   - BETC
      DIS   = DZ   *COSD(PHISC) + DY   *SIND(PHISC)
      FA=ALP - ALPSV
      FB=BET - BETSV
      FD=DIS - DISSV
      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
      F   = FA*DALP   + FB*DBET   + FD*DIS
      DO 100 ICTHE=1,3
      CALL CONV(THESC ,F,THESCSV,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC   ,PHISC,THESC2,PHISC2,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC   ,PHISC,THESC2,
     *          DY   ,DZ   )
      DALP  =ALP   - ALPC
      DBET  =BET   - BETC
      DIS   = DZ   *COSD(PHISC) + DY   *SIND(PHISC)
      F   = FA*DALP   + FB*DBET   + FD*DIS
      IF(ABS(F).LT.TOLF) GO TO 110
  100 CONTINUE
  110 CONTINUE
      IF (THESC .EQ. 0.) THESC=.000001
      PHISCSV=PHISC-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISCSV,THESC2,
     *          DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
```

```
      DISSV = DZSV*COSD(PHISCSV) + DYSV*SIND(PHISCSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC  ,THESC2,PHISC2,ALP  ,BET  )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC  ,THESC2,
     *         DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      FA=ALP-ALPSV
      FB=BET-BETSV
      FD=DIS - DISSV
      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
      F   = FA*DALP   + FB*DBET   + FD*DIS
      DO 200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC  ,THESC2,PHISC2,ALP  ,BET  )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC  ,THESC2,
     *         DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      F   = FA*DALP   + FB*DBET   + FD*DIS
      IF(ABS(F).LT.TOLF) GO TO 210
  200 CONTINUE
  210 CONTINUE
      THESC2S=THESC2-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC,THESC2S,PHISC2,ALPSV,BETSV)
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2S,
     *         DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *         DY  ,DZ  )
```

```
               DALP=ALP-ALPC
               DBET=BET-BETC
               DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
               FA=ALP -ALPSV
               FB=BET -BETSV
               FD=DIS - DISSV
               FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
               F   = FA*DALP   + FB*DBET   + FD*DIS
               DO 300 ICTHE2=1,3
               CALL CONV(THESC2,F,THESC2S,FSV)
               CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *                   THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
               CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *              DY  ,DZ  )
               DALP=ALP-ALPC
               DBET=BET-BETC
               DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
               F   = FA*DALP   + FB*DBET   + FD*DIS
               IF(ABS(F).LT.TOLF) GO TO 310
  300      CONTINUE
  310      CONTINUE

C                CHECK TO SEE IF MOVEMENT IN THESC, PHISC AND THESC2 IS LESS THAN
C                TOLANG IN LAST ITERATION
          IF (  ABS(THESC-THESSAV).LT.TOLANG
     *  .AND.ABS(PHISC-PHISSAV).LT.TOLANG
     *  .AND.ABS(THESC2-THES2SV).LT.TOLANG) GO TO 1010
          THESSAV=THESC
          PHISSAV=PHISC
          THES2SV=THESC2
 1000 CONTINUE
 1010 CONTINUE

             PRINT 97, ICONV
             PRINT 94
             PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ

          IF (IREPT.GE.12) GO TO 3020
```

```
C             CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
      IF(PHISC.GT.PHISLU) THEN
         PHISC=PHISC-180.
         THESC=-THESC
         GO TO 50
      END IF
      IF(PHISC.LT.PHISLL) THEN
         PHISC=PHISC+180.
         THESC=-THESC
         GO TO 50
      END IF
C             SAVE ANSWERS UNLIMITED IN PITCH FOR NEXT COMPUTATION
       THESCUL=THESC
       PHISCUL=PHISC
       THES2UL=THESC2
C             CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT
      IF (THESC.LT.THESLL) THEN
       THESC=THESLL
       GO TO 2020
      END IF
      IF (THESC.GT.THESLU) THEN
       THESC=THESLU
       GO TO 2020
      END IF
         PRINT 98
         PRINT 96
         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
     *           PHIOFF
C          FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
C          WITHOUT REGARD TO DISTANCE (DIS)
      DO 2000 ICONV=1,100
```

```
      THESCSV=THESC-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *              THESCSV,PHISC,THESC2,PHISC2,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *              THESC   ,PHISC,THESC2,PHISC2,ALP   ,BET   )
      DALP  =ALP  - ALPC
      DBET  =BET  - BETC
      FA=ALP - ALPSV
      FB=BET - BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 1100 ICTHE=1,3
      CALL CONV(THESC  ,F,THESCSV,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *              THESC   ,PHISC,THESC2,PHISC2,ALP   ,BET   )
      DALP  =ALP  - ALPC
      DBET  =BET  - BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1110
 1100 CONTINUE
 1110 CONTINUE
      IF(THESC.EQ.0.) THESC=.000001
      PHISCSV=PHISC-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *              THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *              THESC,PHISC   ,THESC2,PHISC2,ALP   ,BET   )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP-ALPSV
      FB=BET-BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
```

```
      DO 1200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC,PHISC  ,THESC2,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1210
 1200 CONTINUE
 1210 CONTINUE
      THESC2S=THESC2-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC,PHISC,THESC2S,PHISC2,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP -ALPSV
      FB=BET -BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 1300 ICTHE2=1,3
      CALL CONV(THESC2,F,THESC2S,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *             THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1310
 1300 CONTINUE
 1310 CONTINUE

C             CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
      IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 2010

 2000 CONTINUE
```

```
 2010 CONTINUE

          PRINT 97, ICONV
          PRINT 94
          PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET

      IF (IREPT.GE.6) GO TO 3020

C                CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT

      IF (THESC.LT.THESLL) THEN
       THESC=THESLL
       GO TO 2020
      END IF
      IF (THESC.GT.THESLU) THEN
       THESC=THESLU
       GO TO 2020
      END IF

C                CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
      IF(PHISC.GT.PHISLU) THEN
         PHISC=PHISC - 180.
         THESC= -THESC
         GO TO 50
      END IF
      IF(PHISC.LT.PHISLL) THEN
         PHISC=PHISC + 180.
         THESC= -THESC
         GO TO 50
      END IF

C                   GO TO END
      GO TO 3020

 2020 CONTINUE
C           FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
C           WITHOUT REGARD TO DISTANCE (DIS), AND WITH THESC SET TO
C           THE LIMITS (THESLL OR THESLU)
```

```
C        SET TO PREVIOUS ANSWERS LIMITED IN THESC
      PHISC = PHISCL
      THESC2 = THESC2L

         PRINT 98
         PRINT 96
         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
     *            PHIOFF

      DO 3000 ICONV=1,100

      THESC2S=THESC2-1.
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC,THESC2S,PHISC2,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP -ALPSV
      FB=BET -BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 2300 ICTHE2=1,3
      CALL CONV(THESC2,F,THESC2S,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC,THESC2 ,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 2310
 2300 CONTINUE
 2310 CONTINUE

      IF(THESC.EQ.0.) THESC=.000001
      PHISCSV=PHISC-1.
```

```
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISCSV,THESC2,PHISC2,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC  ,THESC2,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP-ALPSV
      FB=BET-BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 2200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC,PHISC  ,THESC2,PHISC2,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 2210
 2200 CONTINUE
 2210 CONTINUE
C             CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
      IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 3010

 3000 CONTINUE
 3010 CONTINUE

         PRINT 97, ICONV
         PRINT 94
         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET

      IF (IREPT.GE.6) GO TO 3020

C             CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT AND SAVE ANSWERS
C              WITH PITCH LIMITED TO THESCLL OR THESCLU FOR NEXT COMPUTATION
      IF(PHISC.GT.PHISLU) THEN
         PHISC=PHISC - 180.
```

```
            THESC= -THESC
            PHISCL=PHISC
            THESC2L=THESC2
            GO TO 50
         END IF
         IF(PHISC.LT.PHISLL) THEN
            PHISC=PHISC + 180.
            THESC= -THESC
            PHISCL=PHISC
            THESC2L=THESC2
            GO TO 50
         END IF

C                 SAVE ANSWERS WITH PITCH LIMITED TO THESLL OR THESLU
C                 FOR NEXT COMPUTATION
          PHISCL=PHISC
          THESC2L=THESC2
 3020    CONTINUE
         CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *            DY  ,DZ  )
            PRINT 98
            PRINT 94
            PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
            PRINT 98
            PRINT 98
         STOP
99       FORMAT (6F9.4,2F7.3,F10.5)
98       FORMAT ( )
97       FORMAT(1H ,"ICONV =",I3)
96       FORMAT("   THESC     PHISC   THESC2   PHISC2    ALPC    BETC",
     *        "   PSIOFF  THEOFF    PHIOFF")
95       FORMAT("   THESC     PHISC  THESC2    PHISC2    DALP",
     *        "    DBET      DY     DZ       F")
94       FORMAT("   THESC     PHISC   THESC2   PHISC2     ALP    BET ",
     *        "      DY     DZ")
93       FORMAT (6F9.4,14X,F10.5)
         END
```

```
      SUBROUTINE VEL (U,V,W,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *                THESC,PHISC,THESC2,PHISC2,ALP,BET)
C
C THIS SUBROUTINE CALCULATES THE ANGLE OF ATTACK (ALP) AND ANGLE OF
C SIDESLIP (BET) OF A WIND TUNNEL MODEL.  INPUTS ARE VELOCITY COMPONENTS
C IN THE WIND TUNNEL (U, V, AND W), THE STING OFFSET ANGLES (PSIOFF, THEOFF,
C AND PHIOFF), THE STING BENDING ANGLES (PSISB, THESB, AND PHISB, AND
C THE STING DRIVE ANGLES (THESC, PHISC, THESC2 AND PHISC2).  THE SUBROUTINE
C CALCULATES THE COMPONENTS OF THE FREE STREAM VELOCITY ALONG THE THREE
C AXES OF THE MODEL AFTER EACH ROTATION.  AFTER THE LAST ROTATION, THESE
C VELOCITIES ARE USED TO CALCULATE THE ANGLE OF ATTACK (ALP) AND THE
C ANGLE OF SIDESLIP (BET).
C
      DOR=57.2957795
C              STING PITCH (Y)
      UB=U*COSD(THESC)-W*SIND(THESC)
      VB=V
      WB=W*COSD(THESC)+U*SIND(THESC)
C              STING ROLL (X)
      UC=UB
      VC=VB*COSD(PHISC)-WB*SIND(PHISC)
      WC=WB*COSD(PHISC)+VB*SIND(PHISC)
C              OFFSET YAW (Z)
      UD=UC*COSD(PSIOFF)-VC*SIND(PSIOFF)
      VD=VC*COSD(PSIOFF)+UC*SIND(PSIOFF)
      WD=WC
C              OFFSET PITCH (Y)
      UE=UD*COSD(THEOFF)-WD*SIND(THEOFF)
      VE=VD
      WE=WD*COSD(THEOFF)+UD*SIND(THEOFF)
C              OFFSET ROLL (X)
      UF=UE
      VF=VE*COSD(PHIOFF)-WE*SIND(PHIOFF)
      WF=WE*COSD(PHIOFF)+VE*SIND(PHIOFF)
C              2ND STING PITCH (Y)
      UG=UF*COSD(THESC2)-WF*SIND(THESC2)
```

```
      VG=VF
      WG=WF*COSD(THESC2)+UF*SIND(THESC2)
C              2ND STING ROLL (X)
      UH=UG
      VH=VG*COSD(PHISC2)-WG*SIND(PHISC2)
      WH=WG*COSD(PHISC2)+VG*SIND(PHISC2)
C              STING BENDING IN YAW (Z)
      UI=UH*COSD(PSISB)-VH*SIND(PSISB)
      VI=VH*COSD(PSISB)+UH*SIND(PSISB)
      WI=WH
C              STING BENDING IN PITCH (Y)
      UJ=UI*COSD(THESB)-WI*SIND(THESB)
      VJ=VI
      WJ=WI*COSD(THESB)+UI*SIND(THESB)
C              STING BENDING IN ROLL (X)
      UK=UJ
      VK=VJ*COSD(PHISB)-WJ*SIND(PHISB)
      WK=WJ*COSD(PHISB)+VJ*SIND(PHISB)
C              ALPHA AND BETA
      IF(WK.EQ.0..AND.UK.EQ.0.)UK=.0000001
      ALP=ATAN2(WK,UK)*DOR
      IF(VK.LT.-1.)VK=-1.
      IF(VK.GT.1.) VK=1.
      BET=ASIN(-VK)*DOR
      RETURN
      END
```

```
      SUBROUTINE DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2,
     *                DY,DZ)
C
C      THIS SUBROUTINE CALCULATES THE POSITION OF THE MODEL CENTER
C WITH RESPECT TO THE TUNNEL AXIS SYSTEM.  THE SUBROUTINE STARTS AT
C THE MODEL AND PROCEEDS TO THE TUNNEL AXIS SYSTEM.  THE ROTATIONS
C AND TRANSLATIONS ARE TAKEN IN REVERSE ORDER FROM THE ORDER USED
C IN THE MAIN PROGRAM.  THE STING BENDING ANGLES ARE IGNORED.
C THE ORDER OF ROTATIONS ARE:
C SECOND STING ROLL (PHISC2), SECOND STING PITCH (THESC2),
C OFFSET ROLL (PSIOFF), OFFSET PITCH (THEOFF), OFFSET YAW (PSIOFF),
C STING ROLL (PHISC), AND STING PITCH (THESC).
C             R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
C             X2 IS THE LENGTH OF THE MAIN STING.
C             X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
C
      XA=0.
      YA=0.
      ZA=0.

C         TRANSLATION  (X)  HIGH ALPHA STING
      XB=XA - X3
      YB=YA
      ZB=ZA

C         2ND STING ROLL  (X)
C     NO EFFECT ON X, Y OR Z

C         2ND STING PITCH  (Y)
      XC=XB*COSD(-THESC2)-ZB*SIND(-THESC2)
      YC=YB
      ZC=ZB*COSD(-THESC2)+XB*SIND(-THESC2)

C         TRANSLATION  (X)   MAIN STING
      XD=XC - X2
      YD=YC
```

```
      ZD=ZC

C         OFFSET ROLL  (X)
      XE=XD
      YE=YD*COSD(-PHIOFF)-ZD*SIND(-PHIOFF)
      ZE=ZD*COSD(-PHIOFF)+YD*SIND(-PHIOFF)

C         OFFSET PITCH  (Y)
      XF=XE*COSD(-THEOFF)-ZE*SIND(-THEOFF)
      YF=YE
      ZF=ZE*COSD(-THEOFF)+XE*SIND(-THEOFF)

C         OFFSET YAW  (Z)
      XG=XF*COSD(-PSIOFF)-YF*SIND(-PSIOFF)
      YG=YF*COSD(-PSIOFF)+XF*SIND(-PSIOFF)
      ZG=ZF

C         TRANSLATION (X)   ,ARC SECTOR
      XH=XG + R1
      YH=YG
      ZH=ZG

C         STING  ROLL  (X)
      XI=XH
      YI=YH*COSD(-PHISC)-ZH*SIND(-PHISC)
      ZI=ZH*COSD(-PHISC)+YH*SIND(-PHISC)

C         STING  PITCH  (Y)
      XJ=XI*COSD(-THESC)-ZI*SIND(-THESC)
      YJ=YI
      ZJ=ZI*COSD(-THESC)+XI*SIND(-THESC)

      DY=YJ
      DZ=ZJ

      RETURN
      END
```

```
      SUBROUTINE CONV (X,Y,XSAVE,YSAVE)
C
C THIS SUBROUTINE CALCULATES THE VALUE OF X WHICH MINIMIZES THE VALUE OF Y.
C THE CALLING PROGRAM CALCULATES THE VALUES OF Y FOR EACH VALUE OF X.
C THE NEW VALUE OF X IS DETERMINED BY CALCULATING WHERE A STRAIGHT LINE
C THROUGH THE LAST TWO PREVIOUS PAIR OF POINTS, X AND Y, AND XSAVE AND YSAVE,
C INTERSECT THE X AXIS.  THIS NEW VALUE OF X IS RETURNED TO THE CALLING
C PROGRAM IN THE X PARAMETER.
C
      X1=XSAVE
      Y1=YSAVE
      X2=X
      Y2=Y
      IF(X2.EQ.X1) GO TO 3
      IF(Y2.EQ.Y1) GO TO 4
      XK=(Y2-Y1)/(X2-X1)
      X=X2-Y2/XK
      XA=(X1+X2)/2.
      DX=ABS(X2-X1)
      IF(ABS(X-XA).GT.3.5*DX) X=XA+3.5*SIGN( DX, (X-XA) )
      GO TO 100
3     X=X2-Y2
      GO TO 100
4     X=(X2+X1)/2.
100   XSAVE=X2
      YSAVE=Y2
      IF(ABS(Y1).LT.ABS(Y2)) THEN
       XSAVE=X1
       YSAVE=Y1
      END IF
      RETURN
      END
```

APPENDIX B

COMPUTER LISTING OF PROGRAM STNG2PR

This appendix contains a computer listing of the program STNG2PR which calculates the wind tunnel sting pitch and roll angles and second sting pitch angle required to obtain a desired angle of attack, α, and sideslip, β, and to position the model in the center of the tunnel for a wind tunnel model with on board accelerometers to measure model pitch and/or roll angles . The program accepts accelerometer measurements of model pitch and roll, wind tunnel stream flow angles in two directions and sting offsets and sting bending angles in three directions

```
      PROGRAM STNG2PR (INPUT,OUTPUT)
C THIS PROGRAM CALCULATES THE STING ANGLES FOR THE HIGH ALPHA STING SYSTEM
C TO GET THE DESIRED ALPHA AND BETA AND THE MINIMUM DISTANCE FROM THE
C CENTERLINE OF THE TUNNEL  ( ONLY THE SECOND PITCH ANGLE IS USED, NOT
C THE SECOND ROLL ANGLE)
C THE PROGRAM ACCEPTS INPUTS FROM ACCELEROMETERS ON BOARD THE MODEL TO
C MEASURE THE MODEL PITCH AND ROLL RELATIVE TO GRAVITY.
C
C ANSWERS SAVED FOR NEXT COMPUTATION
C
C THE ORDER OF ROTATIONS ARE : STING PITCH (THESC), STING ROLL (PHISC),
C OFFSET YAW (PSIOFF), OFFSET PITCH (THEOFF), OFFSET ROLL (PHIOFF),
C SECOND STING PITCH (THESC2), SECOND STING ROLL (PHISC2),
C BENDING IN YAW (PSISB), BENDING IN PITCH (THESB), AND BENDING IN ROLL (PHISB)
C
C CODED BY -- JOHN B. PETERSON, JR.   NASA/LARC/AAD/HRNAB   1990
C

C            STING OFFSETS
      PSIOFF=0.
      THEOFF=-20.
      PHIOFF=0.
      PHISC2=0.

C           STING LENGTHS  (SHOULD BE ON THE ORDER OF 57.3 UNITS SO THAT
C                    DISTANCES AND ANGLES ARE OF SAME ORDER OF MAGNITUDE)
C           R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
C           X2 IS THE LENGTH OF THE MAIN STING.
C           X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
C
      R1=122.5
      X2=87.5
      X3=48.918

C            LIMITS

```

```
C               LIMITS ON STING ROLL ANGLE
      PHISLL=-85.
      PHISLU=100.
C FOR INVERTED RUNS USE FOLLOWING LIMITS
C     PHISLL=80.
C     PHISLU=265.

C               LIMITS ON STING PITCH ANGLE (THESE LIMITS ALSO LIMIT
C               THE AVAILABLE BETA TO +19 & -11 UPRIGHT AND -19 & +11 INVERTED)
      THESLL=-11.
      THESLU=19.

C           STREAM FLOW ANGLES
C             POSITIVE FOR FLOW FROM BELOW AND FROM RIGHT
      SWA=.0
      UWA=.0
C               FREE STREAM VELOCITIES (TOTAL VEL. = 1.0)
      U=SQRT(  1./ (1.+(TAND(SWA))**2+(TAND(UWA))**2 )  )
      V=-U*TAND(SWA)
      W= U*TAND(UWA)

C               SET INITIAL VALUES FOR VARIABLES
      THESC = 0.0
      PHISC = 0.0
      THESC2 = 20.
      PHISC2 = 0.0

      THESCUL=0.0
      PHISCUL=0.0
      THES2UL=20.

      PHISCL=0.0
      THESC2L=20.

      THESSAV=999.
      PHISSAV=999.
      THES2SV=999.
```

```
C               TOLERANCES
      TOLF=.00001
      TOLDAB=.00001
      TOLANG=.01

C                        INPUTS TO CONTROL PROGRAM

C               COMMAND ANGLES
      ALPC=40.
      BETC=10.

C               STING DEFLECTIONS
      PSISB=0.0
      THESB=0.0
      PHISB=0.0

C               STING POSITION
      THES=THESC
      PHIS=PHISC
      THES2=THESC2
      PHIS2=PHISC2

C               MODEL PITCH AND ROLL  (SUBROUTINE SIMUST IS USED HERE
C               TO GENERATE VALUES OF MODEL PITCH AND ROLL FOR THIS PROGRAM.
C               ACTUALLY THEY WILL BE MEASURED WITH ACCELEROMETERS ON BOARD
C               THE MODEL AND INPUT INTO THE PROGRAM AS THEMOB AND PHIMOB)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THES   ,PHIS   ,THES2 ,PHIS2,PSIMOB,THEMOB,PHIMOB)
      THEMOB=THEMOB+0.
      PHIMOB=PHIMOB+0.

C                        END OF INPUTS TO CONTROL PROGRAM

C               SET TO PREVIOUS ANSWERS UNLIMITED IN THESC
      THESC  = THESCUL
      PHISC  = PHISCUL
      THESC2 = THES2UL
```

```
C                START OF CONTROL PROGRAM

C         DETERMINE THEORETICAL PITCH AND ROLL OF MODEL
C         AND COMPARE WITH MEASURED PITCH AND ROLL TO GET ERROR

      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THES   ,PHIS   ,THES2 ,PHIS2,PSIMT,THEMT,PHIMT)
      DTHEMOB=THEMOB-THEMT
      DPHIMOB=PHIMOB-PHIMT

************************************************************
C IF THEMOB OR PHIMOB IS NOT AVAILABLE, DTHEMOB OR DPHIMOB
C SHOULD BE SET TO ZERO
C     DTHEMOB=0.
C     DPHIMOB=0.
************************************************************

          PRINT 98
          PRINT 92
          PRINT 99,THEMOB,PHIMOB,DTHEMOB,DPHIMOB

C               SET REPEAT COUNTER TO 0
      IREPT = 0

C                START OVER POINT IF THE PROGRAM CONVERGES OUTSIDE THE
C                THE STING ROLL LIMITS (PHISLL & PHISLU)
   50 CONTINUE

      IREPT = IREPT + 1

         PRINT 98
         PRINT 96
         PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
     *            PHIOFF

C                    CONVERGE ON ALPHA COMMAND (ALPC), BETA COMMAND (BETC),
C                    AND DISTANCE (DIS)
```

```
      DO 1000 ICONV=1,100

      THESCSV=THESC-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESCSV,PHISC   ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESCSV,PHISC,THESC2,
     *          DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC   ,PHISC   ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC   ,PHISC,THESC2,
     *          DY   ,DZ   )
      DALP  =ALP   - ALPC
      DBET  =BET   - BETC
      DIS   = DZ   *COSD(PHISC) + DY   *SIND(PHISC)
      FA=ALP - ALPSV
      FB=BET - BETSV
      FD=DIS - DISSV
      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
      F   = FA*DALP   + FB*DBET   + FD*DIS
      DO 100 ICTHE=1,3
      CALL CONV(THESC ,F,THESCSV,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC   ,PHISC   ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC   ,PHISC,THESC2,
     *          DY   ,DZ   )
```

```
      DALP  =ALP  - ALPC
      DBET  =BET  - BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      F     = FA*DALP   + FB*DBET   + FD*DIS
      IF(ABS(F).LT.TOLF) GO TO 110
  100 CONTINUE
  110 CONTINUE
      IF (THESC .EQ. 0.) THESC=.000001
      PHISCSV=PHISC-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISCSV,THESC2,
     *         DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      DISSV = DZSV*COSD(PHISCSV) + DYSV*SIND(PHISCSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC  ,THESC2,
     *         DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      FA=ALP-ALPSV
      FB=BET-BETSV
      FD=DIS - DISSV
      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
      F   = FA*DALP   + FB*DBET   + FD*DIS
      DO 200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
```

```
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC   ,THESC2,
     *          DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      F    = FA*DALP   + FB*DBET   + FD*DIS
      IF(ABS(F).LT.TOLF) GO TO 210
  200 CONTINUE
  210 CONTINUE
      THESC2S=THESC2-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2S,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2S,
     *          DYSV,DZSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      DISSV = DZSV*COSD(PHISC) + DYSV*SIND(PHISC)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *          DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      FA=ALP -ALPSV
      FB=BET -BETSV
      FD=DIS - DISSV
      FSV = FA*DALPSV + FB*DBETSV + FD*DISSV
      F   = FA*DALP   + FB*DBET   + FD*DIS
```

```
      DO 300 ICTHE2=1,3
      CALL CONV(THESC2,F,THESC2S,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC   ,PHISC   ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP   ,BET   )
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *          DY  ,DZ  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      DIS   = DZ  *COSD(PHISC) + DY  *SIND(PHISC)
      F   = FA*DALP   + FB*DBET   + FD*DIS
      IF(ABS(F).LT.TOLF) GO TO 310
  300 CONTINUE
  310 CONTINUE

C          CHECK TO SEE IF MOVEMENT IN THESC, PHISC AND THESC2 IS LESS THAN
C          TOLANG IN LAST ITERATION
      IF (  ABS(THESC-THESSAV).LT.TOLANG
     * .AND.ABS(PHISC-PHISSAV).LT.TOLANG
     * .AND.ABS(THESC2-THES2SV).LT.TOLANG) GO TO 1010
      THESSAV=THESC
      PHISSAV=PHISC
      THES2SV=THESC2
 1000 CONTINUE
 1010 CONTINUE

      IF (IREPT.GE.12) GO TO 3020

         PRINT 97, ICONV
         PRINT 94
         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ

C             CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
      IF(PHISC.GT.PHISLU) THEN
         PHISC=PHISC-180.
         THESC=-THESC
```

```
            GO TO 50
         END IF
         IF(PHISC.LT.PHISLL) THEN
            PHISC=PHISC+180.
            THESC=-THESC
            GO TO 50
         END IF

C                 SAVE ANSWERS UNLIMITED IN PITCH FOR NEXT COMPUTATION
          THESCUL=THESC
          PHISCUL=PHISC
          THES2UL=THESC2

C                 CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT

         IF (THESC.LT.THESLL) THEN
          THESC=THESLL
          GO TO 2020
         END IF
         IF (THESC.GT.THESLU) THEN
          THESC=THESLU
          GO TO 2020
         END IF

            PRINT 98
            PRINT 96
            PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
        *              PHIOFF

C             FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
C             WITHOUT REGARD TO DISTANCE (DIS)

         DO 2000 ICONV=1,100


         THESCSV=THESC-1.
         CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
        *              THESCSV,PHISC   ,THESC2  ,PHISC2,PSIMT,THEMT,PHIMT)
```

```
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP  =ALP  - ALPC
      DBET  =BET  - BETC
      FA=ALP - ALPSV
      FB=BET - BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 1100 ICTHE=1,3
      CALL CONV(THESC ,F,THESCSV,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP  =ALP  - ALPC
      DBET  =BET  - BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1110
 1100 CONTINUE
 1110 CONTINUE
      IF(THESC.EQ.0.) THESC=.000001
      PHISCSV=PHISC-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
```

```
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP-ALPSV
      FB=BET-BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 1200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1210
 1200 CONTINUE
 1210 CONTINUE
      THESC2S=THESC2-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2S,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
```

```
      DBET=BET-BETC
      FA=ALP -ALPSV
      FB=BET -BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 1300 ICTHE2=1,3
      CALL CONV(THESC2,F,THESC2S,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 1310
 1300 CONTINUE
 1310 CONTINUE

C           CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
      IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 2010

 2000 CONTINUE
 2010 CONTINUE

      IF (IREPT.GE.6) GO TO 3020

         PRINT 97, ICONV
         PRINT 94
         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET

C            CHECK TO SEE IF THESC CONVERGED OUTSIDE THESC LIMIT

      IF (THESC.LT.THESLL) THEN
       THESC=THESLL
       GO TO 2020
      END IF
      IF (THESC.GT.THESLU) THEN
```

```
             THESC=THESLU
             GO TO 2020
           END IF

C                CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT
           IF(PHISC.GT.PHISLU) THEN
              PHISC=PHISC - 180.
              THESC= -THESC
              GO TO 50
           END IF
           IF(PHISC.LT.PHISLL) THEN
              PHISC=PHISC + 180.
              THESC= -THESC
              GO TO 50
           END IF

C                   GO TO END
           GO TO 3020

 2020 CONTINUE
C           FINAL CONVERGENCE ON ALPHA COMMAND (ALPC) AND BETA COMMAND (BETC)
C           WITHOUT REGARD TO DISTANCE (DIS), AND WITH THESC SET TO
C           THE LIMIT (THESLL OR THESLU)

C          SET TO PREVIOUS ANSWERS LIMITED IN THESC
           PHISC = PHISCL
           THESC2 = THESC2L

              PRINT 98
              PRINT 96
              PRINT 99,THESC,PHISC,THESC2,PHISC2,ALPC,BETC,PSIOFF,THEOFF,
     *                 PHIOFF

           DO 3000 ICONV=1,100

           THESC2S=THESC2-1.
           CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *                 THESC   ,PHISC   ,THESC2S,PHISC2,PSIMT,THEMT,PHIMT)
```

```
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
      DBETSV=BETSV-BETC
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP -ALPSV
      FB=BET -BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 2300 ICTHE2=1,3
      CALL CONV(THESC2,F,THESC2S,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 2310
 2300 CONTINUE
 2310 CONTINUE

      IF(THESC.EQ.0.) THESC=.000001
      PHISCSV=PHISC-1.
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISCSV,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALPSV,BETSV)
      DALPSV=ALPSV-ALPC
```

```
      DBETSV=BETSV-BETC
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      FA=ALP-ALPSV
      FB=BET-BETSV
      FSV = FA*DALPSV + FB*DBETSV
      F   = FA*DALP   + FB*DBET
      DO 2200 ICPHI=1,3
      CALL CONV(PHISC,F,PHISCSV,FSV)
      CALL SIMUST(PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *            THESC  ,PHISC  ,THESC2 ,PHISC2,PSIMT,THEMT,PHIMT)
      THEMOBT=THEMT+DTHEMOB
      PHIMOBT=PHIMT+DPHIMOB
      CALL ALPBET(U,V,W,PSIMT,THEMOBT,PHIMOBT,ALP  ,BET  )
      DALP=ALP-ALPC
      DBET=BET-BETC
      F   = FA*DALP   + FB*DBET
      IF(ABS(F).LT.TOLF) GO TO 2210
 2200 CONTINUE
 2210 CONTINUE
C              CHECK TO SEE IF ALP AND BET ARE WITHIN TOLDAB OF ALPC AND BETC
      IF ( (ABS(DALP)+ABS(DBET)) .LT. TOLDAB ) GO TO 3010

 3000 CONTINUE
 3010 CONTINUE

      IF (IREPT.GE.6) GO TO 3020

         PRINT 97, ICONV
         PRINT 94
         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET

C              CHECK TO SEE IF CONVERGED OUTSIDE PHIS LIMIT AND SAVE ANSWERS
```

```
C                WITH PITCH LIMITED TO THESCLL OR THESCLU FOR NEXT COMPUTATION
      IF(PHISC.GT.PHISLU) THEN
         PHISC=PHISC - 180.
         THESC= -THESC
         PHISCL=PHISC
         THESC2L=THESC2
         GO TO 50
      END IF
      IF(PHISC.LT.PHISLL) THEN
         PHISC=PHISC + 180.
         THESC= -THESC
         PHISCL=PHISC
         THESC2L=THESC2
         GO TO 50
      END IF

C                SAVE ANSWERS WITH PITCH LIMITED TO THESLL OR THESLU
C                FOR NEXT COMPUTATION
       PHISCL=PHISC
       THESC2L=THESC2
 3020 CONTINUE
      CALL DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2 ,
     *          DY  ,DZ  )
         PRINT 98
         PRINT 94
         PRINT 99, THESC, PHISC, THESC2, PHISC2,ALP, BET,DY,DZ
         PRINT 98
         PRINT 98
      STOP
99    FORMAT (6F9.4,2F7.3,F10.5)
98    FORMAT ( )
97    FORMAT(1H ,"ICONV =",I3)
96    FORMAT("   THESC     PHISC    THESC2   PHISC2    ALPC     BETC",
     *       "   PSIOFF  THEOFF    PHIOFF")
95    FORMAT("   THESC     PHISC   THESC2    PHISC2    DALP",
     *       "    DBET      DY       DZ       F")
94    FORMAT("   THESC     PHISC    THESC2   PHISC2     ALP     BET ",
     *       "     DY      DZ")
```

```
93    FORMAT (6F9.4,14X,F10.5)
92    FORMAT("   THEMOB   PHIMOB  DTHEMOB  DPHIMOB")
      END
```

```
      SUBROUTINE SIMUST (PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *                   THES,PHIS,THES2,PHIS2,PSIMT,THEMT,PHIMT)

C THIS SUBROUTINE SIMULATES THE STING.  IT CALCULATES THE THEORETICAL MODEL
C YAW, PITCH AND ROLL GIVEN INPUTS OF THE STING OFFSETS, STING BENDING,
C STING ROLL AND PITCH, AND SECOND STING ROLL AND PITCH
      DOR=57.2957795

      CALL COMP (1.,0.,0.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *          THES,PHIS,THES2,PHIS2,XX,YX,ZX)
      CALL COMP (0.,1.,0.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *          THES,PHIS,THES2,PHIS2,XY,YY,ZY)
      CALL COMP (0.,0.,1.,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *          THES,PHIS,THES2,PHIS2,XZ,YZ,ZZ)


C       CALCULATE PITCH OF MODEL
      IF(XZ.LT.-1.)XZ=-1.
      IF(XZ.GT. 1.)XZ= 1.
      THEMT=ASIN(-XZ)*DOR

C           ROLL AND YAW OF THE MODEL
      IF(YZ.EQ.0..AND.ZZ.EQ.0.) THEN
C            CASE WHERE PITCH OF MODEL IS +/-90.
         PHIMT=ATAN2(-YX,ZX)*DOR
         PSIMT=0.
       ELSE
         PHIMT=ATAN2(-YZ,ZZ)*DOR
         PSIMT=ATAN2(-XY,XX)*DOR
      END IF
      RETURN
      END
```

```
      SUBROUTINE COMP    (X,Y,Z,PSIOFF,THEOFF,PHIOFF,PSISB,THESB,PHISB,
     *                    THES,PHIS,THES2,PHIS2,XK,YK,ZK)
C
C THIS SUBROUTINE IS USED TO CALCULATE THE COMPONENTS, IN THE MODEL
C AXIS SYSTEM, OF A VECTOR (X,Y,Z) IN THE WIND TUNNEL AXIS SYSTEM
C
C                STING PITCH (Y)
      XB=X*COSD(THES)-Z*SIND(THES)
      YB=Y
      ZB=Z*COSD(THES)+X*SIND(THES)
C                STING ROLL (X)
      XC=XB
      YC=YB*COSD(PHIS)-ZB*SIND(PHIS)
      ZC=ZB*COSD(PHIS)+YB*SIND(PHIS)
C                OFFSET YAW (Z)
      XD=XC*COSD(PSIOFF)-YC*SIND(PSIOFF)
      YD=YC*COSD(PSIOFF)+XC*SIND(PSIOFF)
      ZD=ZC
C                OFFSET PITCH (Y)
      XE=XD*COSD(THEOFF)-ZD*SIND(THEOFF)
      YE=YD
      ZE=ZD*COSD(THEOFF)+XD*SIND(THEOFF)
C                OFFSET ROLL (X)
      XF=XE
      YF=YE*COSD(PHIOFF)-ZE*SIND(PHIOFF)
      ZF=ZE*COSD(PHIOFF)+YE*SIND(PHIOFF)
C                SECOND STING PITCH (Y)
      XG=XF*COSD(THES2)-ZF*SIND(THES2)
      YG=YF
      ZG=ZF*COSD(THES2)+XF*SIND(THES2)
C                SECOND STING ROLL (X)
      XH=XG
      YH=YG*COSD(PHIS2)-ZG*SIND(PHIS2)
      ZH=ZG*COSD(PHIS2)+YG*SIND(PHIS2)
C                STING BENDING IN YAW (Z)
      XI=XH*COSD(PSISB)-YH*SIND(PSISB)
```

```
      YI=YH*COSD(PSISB)+XH*SIND(PSISB)
      ZI=ZH
C             STING BENDING IN PITCH (Y)
      XJ=XI*COSD(THESB)-ZI*SIND(THESB)
      YJ=YI
      ZJ=ZI*COSD(THESB)+XI*SIND(THESB)
C             STING BENDING IN ROLL (X)
      XK=XJ
      YK=YJ*COSD(PHISB)-ZJ*SIND(PHISB)
      ZK=ZJ*COSD(PHISB)+YJ*SIND(PHISB)
      RETURN
      END
```

```
      SUBROUTINE ALPBET(U,V,W,PSIM,THEM,PHIM,ALP,BET)
C         THIS SUBROUTINE CALCULATES THE ANGLE OF ATTACK (ALP) AND
C ANGLE OF SIDESLIP (BET) OF A WIND TUNNEL MODEL.  THE INPUTS ARE
C THE VELOCITY COMPONENTS IN THE WIND TUNNEL (U, V, AND W), AND THE
C MODEL EULER ROTATION ANGLES, YAW (PSIM), PITCH (THEM) AND ROLL (PHIM).
C THE SUBROUTINE CALCULATES THE COMPONENTS OF THE FREE STREAM VELOCITY
C ALONG THE THREE AXES OF THE MODEL AFTER EACH ROTATION AND THESE VELOCITY
C COMPONENTS ARE USED TO CALCULATE THE ANGLE OF ATTACK AND ANGLE OF SIDESLIP.
C
      DOR=57.2957795
C                        YAW (Z)
      UA=U *COSD(PSIM)-V *SIND(PSIM)
      VA=V *COSD(PSIM)+U *SIND(PSIM)
      WA=W
C                       PITCH (Y)
      UB=UA*COSD(THEM)-WA*SIND(THEM)
      VB=VA
      WB=WA*COSD(THEM)+UA*SIND(THEM)
C                       ROLL (X)
      UC=UB
      VC=VB*COSD(PHIM)-WB*SIND(PHIM)
      WC=WB*COSD(PHIM)+VB*SIND(PHIM)
C                ALPHA AND BETA
      IF(WC.NE.0..AND.UC.NE.0.) THEN
        ALP=ATAN2(WC,UC)*DOR
       ELSE
        ALP=0.
      END IF
      IF(VC.LT.-1.)VC=-1.
      IF(VC.GT. 1.)VC= 1.
      BET=ASIN(-VC)*DOR
      RETURN
      END
```

```
      SUBROUTINE DIST(R1,X2,X3,PSIOFF,THEOFF,PHIOFF,THESC,PHISC,THESC2,
     *               DY,DZ)

C      THIS SUBROUTINE CALCULATES THE POSITION OF THE MODEL CENTER
C WITH RESPECT TO THE TUNNEL AXIS SYSTEM.  THE SUBROUTINE STARTS AT
C THE MODEL AND PROCEEDS TO THE TUNNEL AXIS SYSTEM.  THE ROTATIONS
C AND TRANSLATIONS ARE TAKEN IN REVERSE ORDER FROM THE ORDER USED
C IN THE MAIN PROGRAM.  THE STING BENDING ANGLES ARE IGNORED.
C THE ORDER OF ROTATIONS ARE:
C SECOND STING ROLL (PHISC2), SECOND STING PITCH (THESC2),
C OFFSET ROLL (PSIOFF), OFFSET PITCH (THEOFF), OFFSET YAW (PSIOFF),
C STING ROLL (PHISC), AND STING PITCH (THESC).
C           R1 IS THE RADIUS OF ROTATION OF THE ARC SECTOR.
C           X2 IS THE LENGTH OF THE MAIN STING.
C           X3 IS THE LENGTH OF THE SECOND (ARTICULATED) STING.
C
      XA=0.
      YA=0.
      ZA=0.

C         TRANSLATION  (X)  HIGH ALPHA STING
      XB=XA - X3
      YB=YA
      ZB=ZA

C         2ND STING ROLL  (X)
C     NO EFFECT ON X, Y OR Z

C         2ND STING PITCH  (Y)
      XC=XB*COSD(-THESC2)-ZB*SIND(-THESC2)
      YC=YB
      ZC=ZB*COSD(-THESC2)+XB*SIND(-THESC2)

C         TRANSLATION  (X)   MAIN STING
      XD=XC - X2
      YD=YC
```

```
      ZD=ZC

C         OFFSET ROLL  (X)
      XE=XD
      YE=YD*COSD(-PHIOFF)-ZD*SIND(-PHIOFF)
      ZE=ZD*COSD(-PHIOFF)+YD*SIND(-PHIOFF)

C         OFFSET PITCH  (Y)
      XF=XE*COSD(-THEOFF)-ZE*SIND(-THEOFF)
      YF=YE
      ZF=ZE*COSD(-THEOFF)+XE*SIND(-THEOFF)

C         OFFSET YAW  (Z)
      XG=XF*COSD(-PSIOFF)-YF*SIND(-PSIOFF)
      YG=YF*COSD(-PSIOFF)+XF*SIND(-PSIOFF)
      ZG=ZF

C         TRANSLATION (X)    ,ARC SECTOR
      XH=XG + R1
      YH=YG
      ZH=ZG

C         STING  ROLL  (X)
      XI=XH
      YI=YH*COSD(-PHISC)-ZH*SIND(-PHISC)
      ZI=ZH*COSD(-PHISC)+YH*SIND(-PHISC)

C         STING  PITCH  (Y)
      XJ=XI*COSD(-THESC)-ZI*SIND(-THESC)
      YJ=YI
      ZJ=ZI*COSD(-THESC)+XI*SIND(-THESC)

      DY=YJ
      DZ=ZJ

      RETURN
      END
```

```
      SUBROUTINE CONV (X,Y,XSAVE,YSAVE)
C
C THIS SUBROUTINE CALCULATES THE VALUE OF X WHICH MINIMIZES THE VALUE OF Y.
C THE CALLING PROGRAM CALCULATES THE VALUES OF Y FOR EACH VALUE OF X.
C THE NEW VALUE OF X IS DETERMINED BY CALCULATING WHERE A STRAIGHT LINE
C THROUGH THE LAST TWO PREVIOUS PAIR OF POINTS, X AND Y, AND XSAVE AND YSAVE,
C INTERSECT THE X AXIS.  THIS NEW VALUE OF X IS RETURNED TO THE CALLING
C PROGRAM IN THE X PARAMETER.
C
      X1=XSAVE
      Y1=YSAVE
      X2=X
      Y2=Y
      IF(X2.EQ.X1) GO TO 3
      IF(Y2.EQ.Y1) GO TO 4
      XK=(Y2-Y1)/(X2-X1)
      X=X2-Y2/XK
      XA=(X1+X2)/2.
      DX=ABS(X2-X1)
      IF(ABS(X-XA).GT.3.5*DX) X=XA+3.5*SIGN( DX, (X-XA) )
      GO TO 100
3     X=X2-Y2
      GO TO 100
4     X=(X2+X1)/2.
100   XSAVE=X2
      YSAVE=Y2
      IF(ABS(Y1).LT.ABS(Y2)) THEN
       XSAVE=X1
       YSAVE=Y1
      END IF
      RETURN
      END
```

PSI = ψ = Euler yaw angle = < ABC

THETA = θ = Euler pitch angle = < CBD (Note $\theta \neq \alpha$ unless ϕ = 0°)

PHI = ϕ = Euler roll angle = < CDE

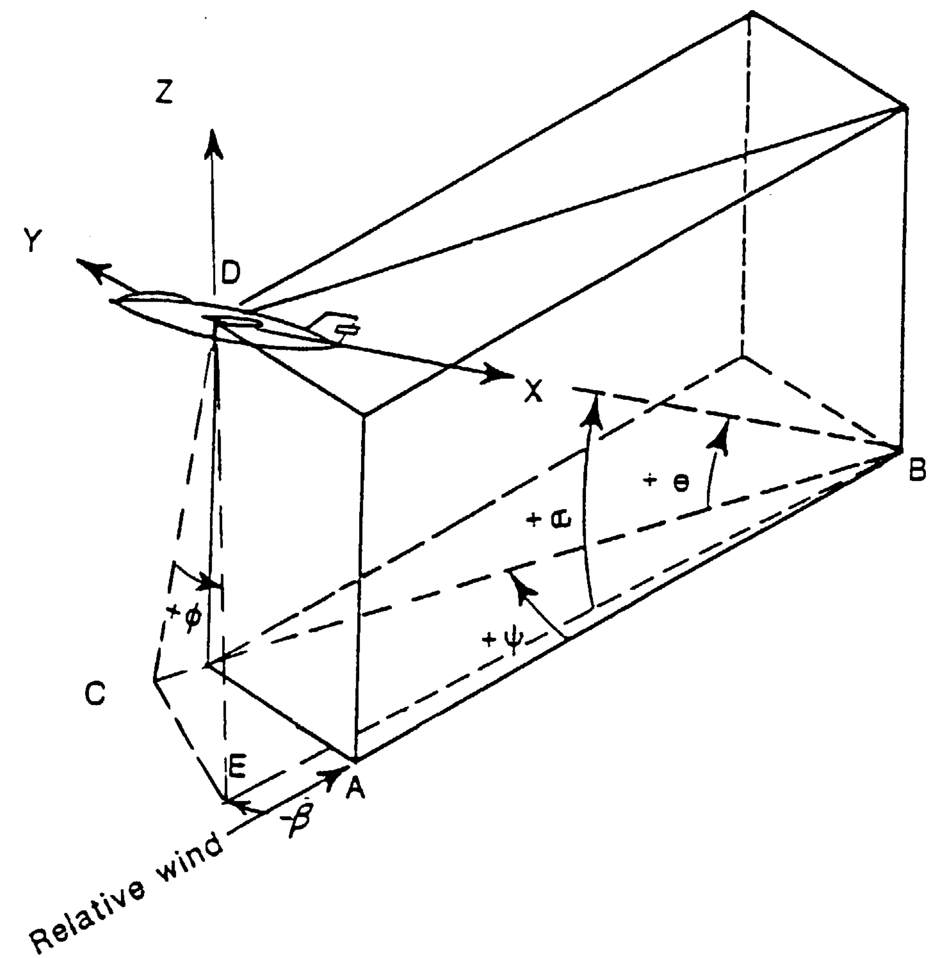

Figure 1. Definition of Euler angles and directions

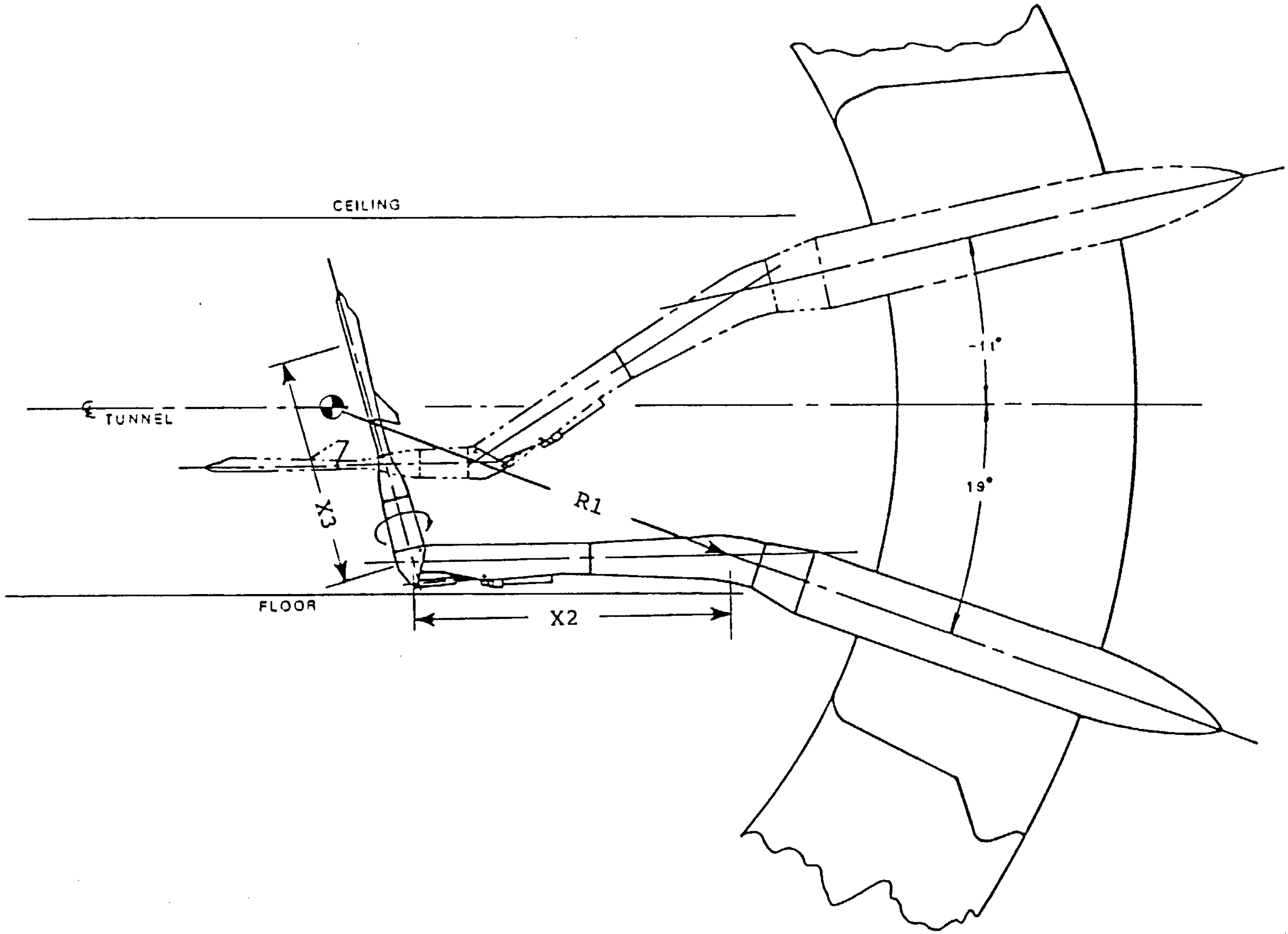

Figure 2. - High angle of attack articulated sting mechanism

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1991

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Computer Programs for the Calculation of Dual Sting Pitch and Roll Angles Required for an Articulated Sting to Obtain Angles of Attack and Sideslip on Wind-Tunnel Models

5. FUNDING NUMBERS
WU 505-59-54-01

6. AUTHOR(S)
John B. Peterson, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM-104161

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-unlimited
Subject category-09

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
Two programs have been developed to calculate the pitch and roll position of the conventional sting drive and the pitch of a high angle articulated sting to position a wind tunnel model at the desired angle of attack and sideslip and position the model as near as possible to the centerline of the tunnel. These programs account for the effects of sting offset angles, sting bending angles, and wind-tunnel stream flow angles. In addition, the second program incorporates inputs from on-board accelerometers that measure model pitch and roll with respect to gravity. The programs are presented in the report and a description of the numerical operation of the programs with a definition of the variables used in the programs is given.

14. SUBJECT TERMS
Wind-tunnel apparatus; Support systems; Computer programs; Angle of attack; Roll; Sideslip; Attitude control; Pitch; Mounts; Yaw; Rotation; and Supports; Struts

15. NUMBER OF PAGES
74

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited